www.freesoftwaremagazine.com

Issue 0 - November, 2004

Free Software MAGAZINE

(TONY MOBILY)

CREATING FREE SOFTWARE MAGAZINE

A COMPLETE AUTOMATED SYSTEM TO CREATE MAGAZINES BASED ON FREE SOFTWARE

(TONY MOBILY)

INTERVEW WITH MYTHIC BEASTS' STAFF

THE SYSTEM ADMINISTRATORS OF MYTHIC BEASTS, PROVIDER OF UNIX SERVERS, EXPLAIN HOW THEY DID IT

(DANIEL JAMES)

THE SECOND MAC REVOLUTION

WHY AND HOW APPLE HAS LEAPED FORWARD THANKS TO FREE SOFTWARE

(DANIEL JAMES)

THE CONTENT TAIL WAGS THE IT DOG

THE CONTENT INDUSTRY IS LOOKING FOR A WORKING BUSINESS MODEL

COVER NOT AVAILABLE

qreate

A PLANT NEEDS WATER TO GROW

Free Software MAGAZINE

BY SUBSCRIBING YOU WILL BE SUPPORTING A MAGAZINE WHICH BELIEVES IN FREE SOFTWARE. ALL OUR ARTICLES ARE RELEASED UNDER THE GNU FREE DOCUMENTATION LICENSE, ENHANCING EXISTING INFORMATION ON FREE SOFTWARE.

SUBSCRIBE NOW!

HTTP://WWW.FREESOFTWAREMAGAZINE.COM/SUBSCRIBE

THE OPEN VOICE NEEDS SUBSCRIBERS TO PROSPER!

Contents

Editorial

What is Free Software Magazine? 5

Tech world

Creating Free Software Magazine 6
by Tony Mobily
A long path that takes us to the very beginning of this project

Case study: Mythic Beasts 12
by Tony Mobily
A small company specialised in Linux servers and amazing support

Word world

The content tail wags the IT dog 15
by Daniel James
Without hardware and software, there would be nothing for digital media to be created on, or used with. And yet the content industry attempts to tell the far larger IT industry what it can and cannot do.

The second Mac revolution 18
by Daniel James
The Mac introduced the creative people in society to computers, changing forever the perception of what these machines could achieve. Could Apple's switch to a UNIX foundation be part of a second radical shift, leaving the PC to play catch-up once again?

A PLANT NEEDS WATER TO GROW

Free Software MAGAZINE

BY SUBSCRIBING YOU WILL BE SUPPORTING A MAGAZINE WHICH BELIEVES IN FREE SOFTWARE. ALL OUR ARTICLES ARE RELEASED UNDER THE GNU FREE DOCUMENTATION LICENSE, ENHANCING EXISTING INFORMATION ON FREE SOFTWARE.

SUBSCRIBE NOW!

HTTP://WWW.FREESOFTWAREMAGAZINE.COM/SUBSCRIBE

THE OPEN VOICE NEEDS SUBSCRIBERS TO PROSPER!

What is Free Software Magazine?

Writing this editorial is much harder than I thought it would be. I made the mistake of leaving it last, and now the problem is that I have written so much over the last three months for this magazine (articles, emails, plans, business letters, personal diaries...) that I can't think of anything I haven't already said at least ten times.

Well, I have to start from somewhere, and I believe I ought to answer the most important question: what is Free Software Magazine?

Allow me to answer by explaining what Free Software Magazine is *not*.

Free Software Magazine is *not* another Linux magazine. There are plenty of them, and they have now reached very high standards. The Open Voice will talk about GNU/Linux *as well as* other important free software which shapes the IT world.

Free Software Magazine is *not* an online magazine. It is printed on real paper, and is sent to its subscribers each month.

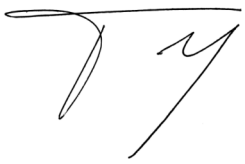
Free Software Magazine is *not* a typical computer magazine. It contains four sections; the first one, *Special*, contains two or three articles about one particular topic; the second one, *Tech world*, is the most technical one and concentrates on case studies and existing solutions. It is not meant to explain the details of how to configure your DNS server or to make your Apache more secure. Rather, this section shows what is achievable with free software, and how. The third section, *Word world*, covers the non-technical side of free software. It contains articles on laws, commentaries on significant episodes, predictions for the future of free software, descriptions of past events, and so on. The fourth section, *Fiction world*, contains short stories that relate to the technological world we live in, and (possibly) free software. I obviously did my homework to find out what people wanted, but I must say that I shaped The Open Voice so that I would buy it if I hadn't been involved in making it.

Free Software Magazine is *not* yet another tiny magazine that will disappear in no time. We are here to stay, to increase our subscriptions and to prosper.

Free Software Magazine is *not quite* a free magazine. All the articles published in The Open Voice (apart from the ones in the *Fiction world* section, for obvious reasons) are released under the GNU Free Documentation License, six weeks after they are published. This means that only subscribers get the latest articles; it also means that over time more and more articles will be available online.

So, what is Free Software Magazine? Well, as it's in your hands (or on your screens) right now, I think I'll take a virtual emergency exit and leave the answer to you. If you find out the answer, don't forget to let me know - along with your comments, ideas and criticisms that will be, I promise, more than welcome.

I will see you here next month!



Free Software Magazine

Free Software Magazine is a magazine by **The Open Company Partners Inc**, 90 Main St. Road Town, Tortola BVI

Editor in Chief
Tony Mobily

Technical Editors
Clare James
Pancrazio De Mauro
Gianluca Insolvibile

Editors
Anna Dymitr Hawkes
Dave Guard

Techs
Gianluca Pignalberi (LaTeX class and magazine generation)
Gian Maria Ricci (RTF to XML converter using VBA)

Graphic design
Alan Sprecacenere (Web, cover and advertising design)
Gianluca Pignalberi, Tony Mobily (Magazine design)

Credits
Michael Eastwood at Firepipe (Initial XSLT transformations and web design)

This project exists thanks to
Donald E. Knuth
Leslie Lamport
People at TUG

For copyright information about the contents of Free Software Magazine, please see the section "Copyright information" at the end of each article. If an article is released under a free license six weeks after publication, and the six weeks are not over yet, you may not reproduce or retransmit the article, in whole or in part, in any manner, without the prior written consent of the author.

Creating Free Software Magazine

A long path that takes us to the very beginning of this project

Tony Mobily

This magazine was inspired by a conversation I had with a great friend of mine called Massimo. I said to Massimo "I think it would be great to start a magazine. It's my ideal job, and I think I know what the world needs right now. It's a pity there's no money in publishing, and I'm not willing to run a magazine that doesn't pay it's contributors *well*...". His answer was very simple: "Tony, there's money everywhere, as long as you do something good and promote it well". Well, seeing that he has a successful business, I thought I would listen. And it's thanks to him that you are reading this article right now.

A few months ago, just before my conversation with Massimo, I realised that the world needed a magazine on free software. My ideal magazine would be aimed at IT professionals (the *techs*) as well as managers.

Massimo said "Tony, there's money everywhere, as long as you do something good and promote it well" Well, seeing that he has a successful business, I thought I would listen

The magazine would contain technical articles, but they would be focused on describing the possibilities of free software, rather than the technical details of how to configure a specific server. It would also publish technical articles on software patents, copyright laws, and how the world is changing thanks to free software. Above all, it wouldn't be yet another technical Linux magazine (there are plenty of them at the moment) and it wouldn't compete with Linux magazines. It would also break the common rules for magazines of this kind, and contain a fiction section - short stories about the new technological era.

This magazine would pay its authors well, and would release all the articles under the GNU Free Documentation License.

A few months ago, after talking to Massimo, I decided that I would do it - and I have.

Starting up

When starting a magazine from scratch, the first concern is creating an appropriate structure to support the project (an office, the staff for the magazine's composition, the managing editor, the web designer, etc).

I have been working on magazines my whole life. I am aware of all the processes involved, and I know that if I had followed the "normal" path, I would have needed a lot of capital to start the project, and a lot of advertisers and subscribers to keep it going. Seeing as Free Software Magazine would attract a restricted audience (we are not Cosmopolitan or Playboy...), such a structure would have been far too expensive.

There is also the technological side of the story. I wasn't willing to accept that Free Software Magazine (I will call it "FSM" from now on) would need a compositor to actually make up the magazine *by hand* for every issue. The manual composition of a magazine and the subsequent quality checks take phenomenal amounts of time and money (for a while I was the man who checked that the all the captions were correct, all the borders aligned, and so on for another magazine). Also, I wasn't willing to accept that in order to create FSM I would need proprietary software (I discovered later that I'd have to give in on this one, but only marginally and temporarily).

A magazine that talks about free software and solutions had to be set up in such a way that composition simply wouldn't be needed: the magazine's PDF and HTML versions would have to be generated *automatically*, providing the articles as input.

Well, I can now say (and not without immense satisfaction, and a sigh of relief): we did it.
The system we created can now take an article written using OpenOffice or Microsoft Word (using the right styles), and generate the XML version - as well as an extremely professional looking PDF file.
It has been an up-hill struggle. I have designed the whole system, coordinated the amazing people who wrote various components, and coded a great deal of it myself. It has been hard. I have sent and received more than two thousand emails for this project. But in the end, we did it. The system is here, and we are now using it for the real "Issue 0" that I am writing right now.

The initial technical planning: XML

A magazine is a collection of articles. Deciding the format for the articles was, in my opinion, one of the most crucial steps of the project - everything else would depend on it. Getting it wrong could have compromised the project's success.
The choice went automatically to XML: it's a language that allows you to define your own file format; it has existing, powerful tools (such as XSLT and XPATH) for converting data into HTML and other formats; it is supported by every platform on this planet; and (which is quite important as well) I had worked on it before, even though it had been a while.
But it's not enough to say, "I'll use XML files", there's still all the design work that needs to be done.
XML lets you decide what tags (or, more correctly, entities) you will use in your mark-up file. You normally do this by writing your DTDs (or more modern schemas). But this decision is an important one, as it's very easy to design an XML structure that simply doesn't work properly. What's worse, you may discover the problem further down the track, when changing a detail in the XML file generates a chain reaction that explodes into many, many hours of work.
This is why the first thing I had to decide was: do I want to use an existing, established DTD, or shall I define my own?
I did my research, which was crippled by my limited knowledge of XML; I looked into other systems that dealt with similar issues, but they all looked too complex or boring to me.
I wanted a simple, lightweight XML structure that would contain exactly what I needed - after all, if XML was a way to store information intelligently, who could decide what information to store better than me? I had worked in the industry long enough to know what information I needed for each article. So, I designed it.

I did my research, which was crippled by my limited knowledge of XML; I looked into other systems that dealt with similar issues, but they all looked too complex or boring to me.

Well, I did it with the help of Michael Eastwood, who has three fantastic qualities: he's a genius, he knows XML very well, and he's a graphic designer. Michael did a lot more than help me with XML: he designed the initial web site, and wrote the XSLT transformations to translate articles into HTML (Michael said from the beginning that it would be up to me to set the XML structure).
Here is what a very basic article looked like:

```
<?xml version="1.0" encoding="UTF-8"?>
<page>

  <title>
    <main>The title</main>
    <sub> The subtitle</sub>
  </title>

  <article_info>
     <publication_date
        day="31"
        month="12"
        year="2004" />

     <authors>
       <author>First Author</author>
       <author>Second Author</author>
     </authors>

     <article_type>Article type</article_type>
     <biography>First Author is this.
                Second Author is that.
     </biography>
   </article_info>

   <contents>

     <p>This is a sample article</p>
     <heading>Today's heading:</heading>
     <p>This is a simple paragraph
        under an heading</p>
```

```
   </contents>
</page>
```

The structure was very simple: everything is enclosed into a <page> tag. An article had a title (divided into main and sub), a publication date, an author, a type, an author's biography, and the text (which was made up of paragraphs).
This is what the resulting HTML looked like after the transformation (please remember that this is a *very* stripped down version of it):

```
<html
<head>

  <title>FSM - The title</title>
</head>

<body>

<p class="article_type">Article type</p>
<br>
<p class="date"> Date: 31/12/2004</p>

<h1>The title</h1>
<h2> The subtitle</h2>

<p class="author">By First Author, Second Author</p>

<p>This is a sample article</p>
<h3>Today's heading:</h3>
<p>This is a simple paragraph under an heading</p>
<h3> About the author </h3>

<p class="biography">First Author is this.
                     Second Author is that.</p>
</body>
</html>
```

I created a directory structure that would contain everything; I placed the HTML documents into `htdocs`, and decided that each directory would only contain one XML file called `index.xml` I also created the directory `bin`, where I placed the command `ov_make_html`. This command basically runs:

```
xmllint --xinclude $OV_PATH/xslt/page_html.xsl \
| xsltproc - index.xml  > index.html
```

The directory tree you can see in the downloads section of this article is a little more complicated (and so are the scripts), but it's easy to find your way once you understand how the whole system works.

Once all this work was done, I candidly asked Michael: "Why don't we use this fantastic system to create the *whole* web site?" Michael was sceptical. He said "you can if you want, but then you are limited: you won't be able to put anything fancy on the web site, only what's allowed in articles".
There is one thing I hate about Michael: he is always right. However, in that particular instance I was right too: using the same template for articles and web pages would simplify the web site's management to a great degree.
What was the difference between a web page and an article? Well, all (and *only*) the information within <article_info> was inappropriate for a web page (author, publication date, article type, and biography). Therefore, a web page would share the same structure as an article, but without the entity <article_info>. This is why both an article and a web page start with <page> (rather than something like <article> or <web_page>).
This was only the beginning. The hard part had yet to come.

Overcoming the limitations: exec filters

In the previous section, I mentioned that using the same converter for articles and for our web site made the latter look really boring.
One of the main technical decisions I made was that even if the site's HTML was going to be generated automatically, the web site itself had to be static. There are several reasons for such a (possibly limiting) choice. A static web site has the following characteristics:

- it can be hosted anywhere;
- it takes less processing power to serve a static pages;
- it is much, much less prone to DOS attacks, SQL injections, and so on;
- it is easier to maintain;
- it will never return an error message to the clients;
- it is much easier to have it mirrored without going insane with mod_rewrite.

So, FSM was going to be an automatically generated, static site.
This created a number of flexibility problems. For example, I wanted every web page to show its "path", so that people visiting it would never get lost while

browsing. I truly dislike web sites that have an unclear and illogical structure (and yes, this does mean that I dislike most web sites on this planet...).

But I wasn't happy to put the path information on the XML file itself: what if the file changed its position in the file system? If the PATH information were in the XML file, I would have to manually change the page's XML as well - unfortunately, I dislike duplication of information as much as illogically structured web sites...

I needed a way of embedding some pseudo-dynamic material in a web page. I also needed to prove Michael wrong, and show him that yes; it was possible to make the web site look great. The basic idea was that the result of a program would be embedded into the resulting HTML page; this would give me a great deal of flexibility.

I didn't find any information on how to do this using "standard" XML. Besides, I needed to be able to:

- run a program which would return XML code, which would then in turn be processed by the XSLT processor (a "pre-processor");
- run a program which would return HTML code (a "post-processor").

This is why I wrote the scripts `ov_pre_exec_filter` and `ov_post_exec_filter` (which are really both the same script, but it changes its behaviour according to its name), which I placed in the `bin` directory of the project.

I wanted to keep the web site "alive": FSM is a magazine, and not a news site; as a result, the only news is when a new issue comes out, and updating the web site only once a month can make it a little "static"

The script `ov_pre_exec_filter` is very simple: it looks for the XML entity `<exec-pre exec="PROGRAM"/>`; it then executes "PROGRAM" and substitutes the program's output with the entity itself. The `ov_post_exec_filter` does exactly the same thing, but it looks for the entity `<exec-post>` instead.

In practice, this would mean for example, that fixing the PATH problem (see above) simply required inserting this entity in the XML file: `<exec-post exec=''ov_exec_path''/>`.

The script `ov_exec_path` would take, as input, the path of the page it refers to, and would print out a complete clickable version of the path.

The exec filters system became much more crucial than I expected. It gave me an incredible amount of freedom, and allowed me to make the web site far less boring.

One fine example is the organisation of the "daily pills". I wanted to keep the web site "alive": FSM is a magazine, and not a news site; as a result, the only news is when a new issue comes out, and updating the web site only once a month can make it a little "static". For this reason, I wanted to have a system where I could publish "daily pills": short articles about absolutely *anything* (even fiction) that could possibly interest the magazine's audience.

This is what the "daily pills" page looks like:

```
<contents>
    <p> Every day (well, nearly every day)
    FSM publishes a <i>pill</i>,
    a short article on Open Source's
    "current affairs".  </p>

    <heading>Index by the year</heading>
     <exec-pre exec="ov_exec_pill_year"/>
</contents>
```

That's all! The script `ov_exec_pill_year` will scan the directory it's in, and give a list of years to choose from. The same applies to `ov_exec_pill_month`, which gives you a list of months, and `ov_exec_pill_day`, which lets you choose what pill you'd like to read.

The exec filters also solved the great problem of the copyright notice, which changes depending on the position of the article (articles in `current_issues` are not yet released as GFDL, GNU Free Documentation License).

So, in the XSLT transformation file `page_html.xsl`, you can see `<exec-post exec=''ov_exec_copyright''/>`, which is expanded into the right copyright notice for that particular article. The script is very smart: if it finds the "LICENSE" file in the directory where the article is placed, then the copyright notice is taken from the "LICENSE" file. This means that I can decide the default copyright (the GFDL if it's in `free_issues`, copy reserved if it's in `current_issues`), or assign

specific copyright for a specific article. This is specifically important for articles in the "fiction" section of the magazine, which are normally not released under the GFDL.

Converting OpenOffice and Word document into XML

After managing a magazine for a few weeks, you discover that it's pretty hard to get an author to follow the writing guidelines.

After a few years, you realise that you were wrong It's not hard... it's *impossible*.

You also realise that very few people want to write their articles using XML tags. This is the reason for authors writing their articles using OpenOffice, StarOffice or Word. The point was: how could we let people write their articles, save them in RTF, and then convert the RTF file into our XML?

There were several answers to this question. The main ones were to:

1. open the RTF file in OpenOffice (in batch mode?), save as OpenOffice's native format (which is XML). Then, apply a XSLT transformation to convert OpenOffice's XML into our own;
2. open the RTF file in OpenOffice or Word, and then use a Basic macro to create the XML file by hand.

The first solution would have been better from a technological point of view, and it would have been a lot faster. The big problem was time. Gian Maria, a Microsoft VBA guru, was available and offered to write the converter - but it had to be a Microsoft Word script.

Writing it wasn't easy at all: we had to decide on a style that would let the authors create a simple, readable RTF document, and convert it into a valid XML file. The solution was (obviously) the use of styles, but we couldn't be too invasive towards the authors, there had to be as few styles as possible, and the RTF document had to be easy to write.

We decided on an optimal file format (which of course was changed a million times in the process) and wrote a finite state automaton that scanned the RTF based on those styles into a valid XML documents.

It would take a long list of articles to explain all the problems we had to solve. In this article, I will only list a few of them:

- API incompatibility between Word for OS X and Winword. I would constantly get back to Gian Maria, saying "you can't really use this function because it's not supported". I don't know how he put up with me...
- UNICODE support. We had to allow UNICODE characters, and the only way to do that was to decode them ourselves (and therefore create the right XML entity). The problem? Think of PCs and Macs... big endian... little endian...
- Speed issues. VBA is very, very slow. The first versions of the converter would take up to 3 minutes to convert a document.
- Text boxes within the articles. The problem with text boxes was that you couldn't use a style for them, because they could contain anything else (headings, bullet lists, etc.).
- Word kept on crashing (especially on Mac).

Gian Maria and I exchanged a ludicrous number of emails. The first document generated by the converter looked OK, and yet it returned about 10 pages worth of problems when checked against the page's DTD. To make things more fun, I would sometimes change the XML format under Gian Maria's nose; some other times, I would report consistency problems in the XML files generated by his macro.

In the end, the converter was rock solid, reasonably fast, reliable on the XML, and above all it *worked*.

I don't think I'll ever be able to thank Gian Maria enough for what he has done He is a busy consultant, and gave up a lot of his free time to write this piece of software.When he offered to write an OpenOffice version of it I simply ran out of ways to say "thank you"! This time, there is no hurry and the XML format is not going to change. That will be the next step and by that point FSM will be entirely based on free software.

The PDF generation

There isn't much I can say about the PDF converter which is responsible for converting a bunch of XML files into a fantastic magazine. Gianluca Pignalberi did it all, and I don't know LaTeX in the slightest!

What I do know, is that he wrote a class called "openpaper.cls", and that his PDF files were simply fantastic. The PDF files created by Gianluca's class don't look like the "typical" LaTeX journals you see around. The PDFs look like a "proper" magazine, with coloured

text, textboxes, and everything else. His LaTeX class creates a fantastic table of contents and a great looking editorial board box.

His work has been invaluable. Without Gianluca, nobody would have ever been able to see a paper version of FSM. In the future, I am sure Gianluca will honour us with a paper on how he created the LaTeX class for FSM.

Without Gianluca, nobody would have ever been able to see a paper version of FSM. In the future, I am sure Gianluca will honour us with a paper on how he created the LaTeX class for FSM.

One thing I can say: I wrote the converter from XML into LaTeX, and realised how hard it is to do it! The problem is that LaTeX wants you to escape particular characters. For example, `$` has to turn into `\$` It would be possible to do so using regular expressions. But you can't! The first reason is that characters like `\` have to turn into `$\backslash$` (which contains itself a `$` symbol...). The second reason is that these rules don't apply in a CDATA section - and I had used CDATA sections for example in listings... This time, it was me who had to write a finite state automaton, in order to deal with all the escaping required by LaTeX. The script is called `ov_latex_preprocess`, and it applies to XML files before they are turned into LaTeX files.

The current problems

So, you may ask: what's missing?

Well, first we have to thoroughly audit our code. In general, the scripts tend to handle problems pretty well, but I am pretty sure I put one or two `die()` statements here and there. I would like to be sure that each script handles critical problems the same way.

Secondly, the system needs to get rid of the need for Microsoft Office to convert RTF documents into XML. This is possible, and it is an absolute priority. Now that we have a working system, and we are in no hurry anymore, we can take the time to "do it right".

We could for example port the macro to OpenOffice and see if we manage to get OpenOffice to process the file in batch mode; or we could write another XSLT transformation to convert OpenOffice's XML into our own.

Thirdly, the system needs a graphical interface. Right now you can only use it by the command line, but the system's inner structure is so consistent that writing a graphical interface would be extremely simple. The possibilities are amazing: you would be able to give your authors a login and a password to access their "slice" of their issue; they could then upload the `index.rtf` file, and see what it will look like once it's online or even printed.

To conclude...

Even though the system has a long way to go before it becomes a fully automated publishing system (if it ever will), FSM *works*. There are still a few bugs to iron out, but in general everything functions.

FSM's system is a little bit like Unix: it's complex as a whole, but each building block is simple and straightforward.

Creating Free Software Magazine gave us an opportunity to work together, and get to know and respect each other more. It is with enthusiasm that we now work on this project, and it will thrive mainly thanks to the hard work we have put in up till now.

Copyright information

©2004 by Tony Mobily

Six weeks after its publication, this article will be released under the license below:

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available at http://www.gnu.org/copyleft/fdl.html.

About the author

Tony Mobily is the Editor in Chief of Free Software Magazine

Case study: Mythic Beasts

A small company specialised in Linux servers and amazing support

Tony Mobily

Mythic Beasts is a UK company that provides Unix shells to their users. They offer fantastic service to people who need a shell account on a very fast server, and don't want to fork out silly amounts of money. Let's talk to Chris Lightfoot, one of the company's owners.

TM: Who is behind "Mythic Beasts"? How did everything start?
There are six of us involved: Paul (see the photo), Richard, Pete, Tom, James and myself. Back in February 2000, I had the idea of setting up a colocated machine for our own use. I've dug out the original mail I sent inviting the others to get involved; it began,

- Plan: to host a computer somewhere, in order to have a decent platform to do web hosting, email, etc.
- Problem: cost.

At that time colocation was a bit more expensive than it is now! The first server itself cost us about £500. Once we'd set things up for ourselves and our friends, we discovered that there was quite a lot of demand for Linux shell accounts, so (in August 2000) we registered as a UK limited company and Mythic Beasts Ltd. was born.

TM: Why the name "Mythic Beasts"?
The name "Mythic Beasts" is a complete accident. Our first server was called "mythic", because my convention is to name machines after adjectives and I thought the word "mythic" sounded good. That suggested the domain name "beasts"; when we came to set up the company - and remember that this was the tail-end of the dot.com era and lots of companies had *much* sillier names - we just recycled the name.

TM: Do you all work in the same office? Or do you normally work from home?
No, we work all over the place, mostly communicating by email and now IRC.

TM: Did you have a specific philosophy behind your idea? Did you decide your target market before hand?
The way the company started was, as I've said, a bit of an accident - we spotted demand for something and were in a position to supply it. The reason we originally set up the services for our own use was that we'd all had bad experiences with shared hosting providers before, and wanted something better.

The way the company started was a bit of an accident - we spotted demand for something and were in a position to supply it.

We don't have an official company philosophy or a mission statement which we engrave on paperweights (we don't even have a pointy-haired boss!) but I think the closest thing we can come up with is that we're all interested in selling services which we wouldn't be ashamed to use ourselves.

TM: What operating system (or systems?) do you run on those servers? How many people actively work on the server for maintenance?
At the moment everything runs Linux, mostly Debian but with some older (and heavily customised) RedHat installs. We're not zealots about operating systems (well, not much, anyway). But Linux does everything we need at the moment.
Most of the server maintenance is done by Paul, Pete and me, but unless we're installing new hardware, there isn't all that much to do. Modern PC hardware seems

Fig. 1: Chris Lightfoot

to be pretty reliable as long as (a) you pay enough for it to start with, and (b) you can hot-swap hard disks in your RAID arrays, since those are the components that fail most often.

TM: Who is your average customer? Do you ever have problems such as people getting an account to install an IRC bot, or to crack other servers?

Our typical shell account customer is interested in hosting one or several websites; we offer more flexibility than most hosting providers - for instance, our customers can write CGI scripts in more-or-less any language they want, can run web sites under FastCGI, etc. - and that's attractive for people who want a service which isn't going to get in their way too much.

We also have quite a few customers who use our services as an email service. Having a reliable shell account where you can run mutt or pine or whatever from anywhere in the world is a lot more convenient than running a desktop mail client and using web mail away from home (though obviously we support both of those options too). We also host some software developers who use our anonymous CVS services.

For users who want more sophistication we now offer a "virtual dedicated server" option, which is based on user-mode Linux; that's been quite popular for customers who want even more choice in what they can do, but don't want the hassle of owning and running their own hardware.

We've had occasional trouble with people buying accounts for IRC bots (which we prohibit, because this activity can be disruptive to our other customers), and also occasional individuals who try to buy accounts with stolen credit card numbers (apparently because this is a way for them to verify the numbers they've got). It's almost always easy to spot these kinds of account applications ahead of time.

TM: How do you organise your customer support?

Mail to support gets dropped into a home-brew ticketing system written by Paul (we used to use RT, but it was slow and overcomplicated). From there mail is distributed to a support list, which currently consists of five of the original investors. The ticketing system keeps track of who "owns" each support issue, and reminds us every morning about which tickets are still open.

TM: Is there a lot of custom software involved?

Yes, quite a bit, mostly written by Paul and me. You can see some of it on our software page at http://www.mythic-beasts.com/software.html (`http://www.mythic-beasts.com/software.html`) and more on our personal home pages. Some of the software we've written has found quite wide use: my POP3 server, tpop3d, is used by quite a few sites (some much larger than Mythic Beasts Ltd.), along with our virtual mail configuration tool, vmail-sql. Both of those were written back in 2000 because there weren't any suitable alternatives.

Paul wrote an excellent little tool called iftop for monitoring network performance in real-time, which I think is now in Debian and the *BSD ports.

TM: Do you release everything you wrote for Mythic Beasts as free software?

We don't always have the time to release things we've written - especially when they're very specific to our setup and therefore not likely to be of general interest.

Fig. 2: Paul Warren

TM: How important is free software for your company?

Companies like Mythic Beasts (not to say the internet as a whole) are only really viable because so much Free software is available, and we're glad to contribute what we can to the community.

TM: How much "politics" is involved in running a service like yours? For example, are you criticised for not having any Windows machines? Or for only having Linux? Have you lost any customers because of the SCO legal challenges?

We've encountered surprisingly little politics in running Mythic Beasts. As most of our users are familiar with Linux, and often have a strong preference for it, it is quite unusual for us to get requests for other operating systems. Almost all the services we run are based on open standards in any case, so we don't put any barriers in the way of people who use Windows on the desktop (which is probably the majority of our customer base).

We do have colocation customers who use a variety of operating systems.

As to SCO's recent legal challenges, we have not lost any customers due to this, and, if anything, have had more new customers since the challenges. We do not presently regard the whole SCO fiasco as a threat to our business (or to anyone else who uses Linux).

TM: What systems do you have in place for backup and disaster recovery?

On-site and off-site backups. All our customer-facing servers are hot-swap hardware RAID, backed by a support contract with our suppliers. The VDS servers back each other up, allowing us to restore a VDS on a different host within a few minutes - much better than most dedicated server arrangements.

TM: What are your plans for the future? I see that you are expanding your offer. Are you planning on growing considerably in the medium and long term?

Our plan is for steady growth with customer demand rather than significant expansion. We've invested a fair amount in new hardware this year and now have plenty of capacity for new customers.

We launched our VDS service which is proving very popular for a wide variety of applications that can't be handled easily in a shared hosting environment such as Java servlets and Subversion repositories, and for people who just want absolute control over the hosting of their website and email.

TM: Thank you very much for answering my questions!

Copyright information

©2004 by Tony Mobily
Six weeks after its publication, this article will be released under the license below:
Verbatim copying and distribution of this entire article is permitted in any medium without royalty provided this notice is preserved.

About the author

Tony Mobily is the Editor in Chief of Free Software Magazine

The content tail wags the IT dog

Without hardware and software, there would be nothing for digital media to be created on, or used with. And yet the content industry attempts to tell the far larger IT industry what it can and cannot do.

Daniel James

The content industries have conspicuously failed to create a business model based on paid content over public IP networks, but still cling to the idea that those networks were created for just that use. Any software or system which might interfere with this theoretical paid content business is considered not just heretical, but probably criminal. The music and movie consortia have turned the transition to network distribution into a "with us or against us" battleground, with most of their customers fighting for the wrong side.

RIAA, copyright and file sharing

In a networked world, we would be lost if we couldn't find data on other computers. But four college students in the US have recently been made painfully aware that the mere act of searching and indexing networks can now be considered 'contributory copyright infringement'.

In a networked world, we would be lost if we couldn't find data on other computers. But four college students in the US have recently been made painfully aware that the mere act of searching and indexing networks can now be considered 'contributory copyright infringement'

Without warning, they were sued by the Recording Industry Association of America for damages of up to US$150,000 per song discovered on the campus LAN. Their crime was to have hosted search tools which indexed the files made available to guest users by fellow students - whether those files were music, or college coursework.

Of course, they weren't the only people running search engines on their respective campuses - any copy of Microsoft Windows includes SMB search tools which can perform the same function. The prosecution of these four individuals seems designed to create an example; to let the general public know that the content industry will go to any lengths, no matter how vindictive, unpopular or impractical, to defend the paid content dream.

Faced with a potentially bankrupting legal action, the students settled out of court for between US$12,000 and $17,000 each. Some of the students have asserted that they still don't believe they have done anything wrong, while being constrained by the terms of the settlements as to what they can say in public. Although these sums are tiny compared to the original damages claims and would hardly cover the RIAA's legal fees, they represent the college savings of these four people. The RIAA has indicated that it will be pursuing more individuals for damages, and that the sums it is prepared to settle for will only get higher.

Never mind that an index of SMB shares on a particular LAN is an equally useful aid to institutions attempting to discover copyright violations as it is to the freshman who wants to hear the new Madonna album before he buys it. Never mind that the RIAA has exploited technological ignorance in its attempt to equivocate these college search engines with Napster, which was supposed to be a profit making company.

By extracting multi-thousand dollar tributes from individuals who haven't made a penny from the sharing of

Textbox 1: Get your money for nothing...

Faced with a failing business model? Can't keep up with technology, or shift the units like you used to? If you've got nothing left to lose, why not try the secondhand ideas business?
The SCO Group (the company formerly known as Caldera, not the original proprietary-UNIX-on-Intel Santa Cruz Operation) recently compared Linux developers and users with peer-to-peer music sharers. In the infamous "Letter to Linux Customers" - not that SCO ever had very many Linux customers - CEO Darl McBride wrote: "Similar to analogous efforts underway in the music industry, we are prepared to take all actions necessary to stop the ongoing violation of our intellectual property or other rights."
In fact, SCO/Caldera was doing much worse business than most of the music industry until it decided to make source code licensing a mainstay of its revenue stream. The company was losing millions of dollars each quarter like the dot-com operation it is - until the SCOsource initiative was launched. Now it has reported a net income of US$4.5 million for the quarter to April 30, 2003 on revenue of $21.4 million. SCOsource was said to be responsible for $8.3 million of that revenue.
This model of licensing inherited 'intellectual property' is also a potential saviour for the recording industry. If actually producing and distributing new music is no longer profitable enough to interest the major labels, they could simply re-hash old material - by starting a 1980's revival, for instance. In this analogy, the SCO source code licensed from Novell would be the back catalogue of Dire Straits; very popular in its day, but few people would admit to keeping a copy around now.
Having first made sure that the copyright signed over by artists was extended for long enough to extend past the furthest profit forecasts - say, the artist's death plus seventy years - the same old rubbish could be recycled for each new generation of music consumers. Why hasn't the music industry thought of this already?

copyrighted work, the RIAA has blown apart its rationale for legal action - that of the artist as the victim. It provides evidence that the content industry really does feel more threatened by new forms of network distribution than it does by illegal counterfeit CD and DVD pressing plants. We haven't yet seen individuals given punitive fines for advertising car boot sales, or signposting street markets where counterfeit merchandise might be available.

Where are the thieves?

Where might the principle of "contributory infringement" end? Once again, as in the deCSS cases brought by Hollywood, exercising the right to create a link from one machine to another across a network has suffered from legal intimidation. An open IT system has the potential for as many abuses of copyright as it has for creativity, not just within the operating system but from the hardware to the network and the unfettered blank media. But we probably won't see the RIAA take the IT companies, telcos and consumer electronics manufacturers who make file sharing possible to court.
In the essay Content is Not King, Andrew Odlyzko pointed out that only a minority of the value of public networks consisted of copyrighted, professionally produced 'content'. If we attempt to illustrate this point by comparing annual revenues across the industries, we discover there are probably several IT companies which each have greater annual revenue than the entire global music industry.

Once again, as in the deCSS cases brought by Hollywood, exercising the right to create a link from one machine to another across a network has suffered from legal intimidation.

According to the International Federation of the Phonographic Industry, the international counterpart to the RIAA, recorded music sales worldwide were down 7% to US$32 billion in 2002. Gartner estimates that the global expenditure on IT products and services in the same year was also down, which might suggest economic conditions rather than nefarious students as

the cause of music industry woes. Including telecommunications, the figure for global IT spend in 2002 was $1.521 trillion, of which $556 billion was for IT services alone.

> The Soviet Union managed to control the creative technologies to the extent that dissidents had to make 'samizdat' copies one at a time, by hand

The global media companies which make up the content industry have a well organised publicity and lobbying machine, which has so far been able to lead the public debate on how media and IT networks should converge. Accordingly, people who listen to music on computers are thieves, while those who listen to the same music on radios are true fans. That there is less and less difference between computers and radios seems not to matter. The spin on IT perpetuated by the content industry has also lead to statements that every CD-R sold represents a lost album sale. Have these people not heard of data backups?

Striking back

Every time a music or film industry spokesperson appears in the media to call for tighter controls on technology, there should be a rapid rebuttal from a technologist. When the vested interests of the content industry claim to be defending the artist in front of a political committee, there should be someone there sticking up for the long-established rights of everyone else.

> It's up to the IT industry to educate the public and politicians of the real value of open systems.

The Soviet Union managed to control the creative technologies to the extent that dissidents had to make "samizdat" copies one at a time, by hand - turning the distribution of ideas back to what it had been before Gutenberg. Given the opportunity, the content industry would love to limit powerful and versatile general-purpose computing to selected and approved individuals who could be trusted to respect the rules laid down by the owners of culture. It may seem an extreme comparison, but both the Soviet regime and the content industry have attempted to outlaw unencumbered copying machines.

It's up to the IT industry to educate the public and politicians of the real value of open systems. The music business is small compared to the industry it seeks to dominate. And yet, if it can get enough political support, it may end up doing just that.

Bibliography

[1] Content is Not King, http://firstmonday.org/issues/issue6_2/odlyzko/

[2] Global sales of recorded music down 7% in 2002, http://www.ifpi.org/site-content/statistics/worldsales.html

[3] 2002 Worldwide Hardware and Software Product Support Market Size and Forecast, http://www4.gartner.com/DisplayDocument?id=383387

[4] An analysis of the RIAA's complaint against Daniel Peng, http://barillari.org/papers/peng/peng.html

[5] Daniel Peng's site, http://m-net.arbornet.org/~danpeng/

[6] Jesse Jordan's site, http://www.chewplastic.com/

Copyright information

©2004 by Daniel James

Here are the article's license, effective now:

You may not reproduce or retransmit the materials, in whole or in part, in any manner, without the prior written consent of the author.

About the author

Daniel James is a freelance technology writer.

The second Mac revolution

The Mac introduced the creative people in society to computers, changing forever the perception of what these machines could achieve. Could Apple's switch to a UNIX foundation be part of a second radical shift, leaving the PC to play catch-up once again?

Daniel James

The Apple machines of the 80's turned computers from dull counting machines and glorified typewriters into creative tools, forcing a revolution in the design, publishing and music industries through accessibility and mass participation. If you wanted to produce - let's say - a magazine in those days, all of a sudden you didn't need a lot of money, you didn't need to have a particular job or be a member of the union, and you didn't even need to have served an apprenticeship or have been to college.

Participation is based on ability, not who you are or where you come from. If you are talented enough, you can join a peer group of the world's best programmers and get respect for your contribution.

The new tools were good enough that talented people could figure them out for themselves, and a creative explosion was the result.

Just as the Mac opened up those businesses to a new generation, now open source software (or in the terms some people prefer, "free" or "libre" software) is doing the same for the development of computer systems. Participation is based on ability, not who you are or where you come from. If you are talented enough, you can join a peer group of the world's best programmers and get respect for your contribution - even though you may be just a teenager from some remote land without a penny to your name. The converse is also true; you may have looks, money and a massive ego, but without a clear contribution to make you will probably be ignored. Free software is a harsh but ultimately fair meritocracy.

The great advantage

Given that the Mac platform is now taking advantage of this mass participation, by moving to a UNIX family kernel and picking up the source code of software like KHTML to create Safari, there's a resonance with the GUI computing revolution of the 80's. OS X can run XFree86, the version of X11 used on nearly all Linux desktop systems. The cross-pollination between OS X, UNIX and Linux means that Apple users, for the first time, can participate in the development of their operating system of choice at the deepest level.

There's a positive feedback loop between users and developers which can only improve the software that we rely on. It's not by accident that UNIX family systems have a reputation for stability and efficiency - it's a direct consequence of active input from countless users over thirty years and more. When there's a problem or someone figures out an improvement, users with source code don't just moan or sit waiting for an upgrade - they get involved. And having fixed the problem, they share the information so other users can fix their systems.

This participation is denied to Windows users, who are stuck in the one-way supplier to customer relationship created by the proprietary approach to software. Even if they wanted to understand Windows, fix or improve it, they aren't allowed to. Windows source code isn't generally available to individuals, and is only rarely

available to institutions - and then under very restrictive terms.

So the previously untapped creative potential of the world's computer users has been unleashed on free software - just look at the statistics of the http://sourceforge.net site for an example. Sourceforge is the largest site hosting open source projects, offering server space and collaborative tools. It currently hosts over 71,000 individual software development projects, with more than ten times that number of registered users.

No single company can employ enough software developers to compete with these massive collaborative projects - not even Microsoft. To give just one example, 2003 has seen the launch of the 64-bit Apple G5 hardware, the semi 64-bit Panther and new Linux distributions for the AMD64 CPUs. Windows users can buy a 64-bit Opteron or Athlon64 chip from AMD today, but they can only run a 32-bit legacy operating system until at least next year - so much for the supposed technical lead of Microsoft. But then they were late to the table with the graphical desktop too.

Catching those 64 bits

UNIX family operating systems have several advantages in the migration to 64-bit platforms, not least that the technology has been around for quite a while. Linux has been available for the 64-bit Alpha CPU since 1995, and some of the proprietary UNIX variants have been 64-bit for much longer. But the key advantage over Windows is in source code availability. You can't port to 64-bit without it, and if source code is distributed among many thousands of users and developers then that makes the considerable effort of porting and testing that much more likely to happen.

UNIX family operating systems have several advantages in the migration to 64-bit platforms, not least that the technology has been around for quite a while.

When Microsoft finally has its 64-bit Windows ready for launch, it will have to go to each and every significant application developer and ask them to commit resources to developing a 64-bit version of their product. The answer is likely to be that the application companies will wait and see if 64-bit Windows becomes popular - a chicken and egg problem if ever there was one.

UNIX family systems already have the free software GCC compiler, which as of version 3.3 supports the optimisations which allow programs to take advantage of 64-bit features on the G5 platform. Other critical tools and libraries are well established, thanks in part to IBM's use of a 64-bit PowerPC CPU, known as the POWER4+, in its pSeries server range.

Conclusions

But that talented and yet unknown teenager working on the next killer application is hardly likely to have access to a machine like an IBM pSeries.

Apple's G5 desktop machines are built and priced for the mass market, and so along with machines based on the new AMD chips, we'll see them adopted by the next generation of 64-bit developers.

They can cost tens of thousands of pounds, and are usually only found in enterprise IT departments. By contrast, Apple's G5 desktop machines are built and priced for the mass market, and so along with machines based on the new AMD chips, we'll see them adopted by the next generation of 64-bit developers. Given that all the tools are in place and plenty of source code is available to anyone who wants it, tomorrow's systems will be likely to be created with free software.

Copyright information

© 2004 by Daniel James
Here are the article's license, effective now:
You may not reproduce or retransmit the materials, in whole or in part, in any manner, without the prior written consent of the author.

About the author

Daniel James is a freelance technology writer

A PLANT NEEDS WATER TO GROW

Free Software MAGAZINE

BY SUBSCRIBING YOU WILL BE SUPPORTING A MAGAZINE WHICH BELIEVES IN FREE SOFTWARE. ALL OUR ARTICLES ARE RELEASED UNDER THE GNU FREE DOCUMENTATION LICENSE, ENHANCING EXISTING INFORMATION ON FREE SOFTWARE.

SUBSCRIBE NOW!

HTTP://WWW.FREESOFTWAREMAGAZINE.COM/SUBSCRIBE

THE OPEN VOICE NEEDS SUBSCRIBERS TO PROSPER!